



Document

BlackBerry Internet Content Functional Specification

Page

1(16)

Author Data Gary Mousseau	Date	Document No. Document 990422-001
Approved	Rev Rev 1.3	File Reference Internet Content

BlackBerry Internet Content Functional Specification

Prepared By: Gary Mousseau



Table Of Contents

1	Introduction.....	3
2	Terminology.....	3
3	User Experience.....	4
4	Complexity Questions.....	5
5	Functional Details.....	7
5.1	Message Viewer.....	7
6	Internet Browser.....	8
6.1	Traditional Internet Browser.....	8
6.2	Program Interpreter.....	8
7	Host Development.....	11
7.1	RIM SWS Requirements.....	11
7.2	Host Filtering Requirements.....	12
7.3	Blackberry Handheld Filtering.....	12
7.4	Blackberry Filtering Commands.....	14
8	Content Provider.....	15
8.1	Native HTML Content.....	15
8.2	Native RIM Content.....	15
9	Conclusions.....	16



1 Introduction

The Blackberry solution requires a method for accessing Internet content to solve a wide variety of requirements and problems. This document's goal is to identify these problems and discuss how they will be solved. This document has been through many revisions and is meant to reflect an evolving view of the final product. The solutions presented in this specification represent a novel way to solve the Internet Content problem; this document does not just describe a simple browser on the pager solution.

2 Terminology

Most readers probably know general terms that will be used in this document, but some terms do need to be defined.

- **HTML** : Hypertext Markup Language is the basis for content delivery in the World Wide Web today. This standard will continue to be persistent for sometime to come and must be a fundamental part of the solution being proposed for Blackberry.
- **Gizmos** : <INSERT MARKETING TERM HERE!> This is a throw away word that the marketing department needs to replace. The term required must represent an aggregate of ideas that represents either (i) a URL in the cases where the information is HTML centric, (ii) a File where the information is a program, (iii) a Form where the information is a Java Script or a WML Script or (iv) a deck where the information is an HDML or WML deck of cards. Such terms as 'Item', 'Link', 'File', 'Program', or 'Active_Item' may all be useful terms to use.
- **Sections**: Is a term used in connection with HTML web pages within this document to represent a portion or fraction of the enter page. The term section could also be used in conjunction with WML or JAVA pages if they were also being delivered and categorized to the device.
- **URL** : The term URL stands for Universal Resource Locators and is well know to most of the readers. This term may also be used in conjunction with the term Gizmos as an aggregate term for several types of data.
- **WAP** : Wireless Access Protocols – is a collection of protocols that are being standardized for delivering information to handheld devices.
- **WML** : Wireless Markup Language – is an XML-based markup format used to deliver content in a WAP environment. WML operates with its own XML dictionary and data tag definitions to deliver cards and decks to handheld devices.



Date	Rev Rev 1.3	Document No Document 990422-001
------	----------------	------------------------------------

3 User Experience

How will this new facility in Blackberry be used and experienced? This question must be answered very specifically and completely if RIM is to build a product that actually solves the right problems. Due to the complexity and challenges in solving the Internet content problem RIM must select a subset of problems to solve, and a subset of complexity to undertake. The user cases that are foreseen include:

1. A user receives an e-mail message with HTTP references embedded within the message. The user should be allowed to roll to these references and perform a fetch of this URL from a browser like interface. Once this information is filtered for Text-only, or other formatting information supported by the handheld, it is delivered directly to the browser on the pager.
2. The user has the ability to interact with a series of forms; either received from their company by preloading the forms, getting them via e-mail or by directly fetching them from the browser interface. These forms are either Internet or Intranet based forms, and what exactly is in these forms the user does not necessarily care. The user can select any one of these forms load them into a browser like interface, fill in all the required fields, and return a response to the original web site or Intranet site.
3. The user can invoke the browser directly from the Blackberry Ribbon. They can browse through a cache of locally saved files, like Windows Explorer, or even select a completely random file (URL) to download or fetch. In many cases a previously saved Bookmark (URL) would be a form that can be filled in and submitted to the destination URL address.
4. A host or server developer can follow a document and create a script-based program (or Java) that can be downloaded to the Blackberry pager. The program can display forms and perform local service calls via the O/S and loaded applications, UI32 or the Ribbon application. The downloaded script can check and verify user input and perform a wide range of local decision paths. The types of languages RIM chooses to support is completely up to what has been built into the proxy server for translation. It should also be possible to send a 'RIM-Native' script to the device that would require virtually no pre-processing. This would allow encryption and compression to be preformed before it reaches the proxy server for delivery.



4 Complexity Questions

With each of the four user cases just presented, there is a range of complexity that can be undertaken to achieve the desired results. How far should RIM take each of the four user experiences listed above? It is critical to answer this question to ensure that the time required building the product hit the marketing window for when the product is needed.

General Browser Options

- Limiting all incoming e-mail to contain a link only and no HTML or tokenized scripting information will save work and keep the two components very separate. It might be possible to included an HTML attachment along with the clear text message that the user can open to spawn the browser.
- Does the user require the ability to attach a URL reference into an outgoing mail message? This could either be supported by using a copy/paste method through the UI engine, or alternatively by allowing a 'Send' menu option within the browser.
- The caching model (management) for handling URLs could become very complex. It is the development team's initial expectation that the cache will work similar to the message list. Oldest, unused items will be dropped off the list and deleted.

Advanced Forms and Scripting

- The ability to support forms and user input strongly implies that we must support a scripting language. The scripting language is optional however and RIM could simply use tokenized HTML forms for the presentation. The disadvantage by taking this 'limited' method is that 'interactions' with the server are likely to increase; matching HTML interactions with a normal web server. Each field entered by the user could be incorrect, or spawn other sub-fields which could take valuable time and battery power. This limited model also works very poorly as coverage varies and message can't leave the pager in a timely fashion.
- As part of a company building a vertical solution the company should be able to pre-load the Blackberry Handheld unit with any necessary forms and information that will be used with the browser. Pre-loading information on the device is akin to pre-loading the cache over the serial port to save time and battery power.
- While in the browser the user can save URLs (bookmarks) and recall them. Saving a URL may also result in that URL being placed into the cache, in the Netscape or IE world this is normally not the case. However, in the Blackberry Handheld case every attempt must be made to reduce over-the-air traffic. In many cases it is expected that saved URLs would include forms and other standard documents needed by the travelling executive on the road.

Advanced Form Alternatives

- The user or company developer desires the ability to develop custom programs to be downloaded to the Blackberry Handheld. Another way to achieve this functionality is to send down programs or applets, like a Java applet, that would support a



Date	Rev Rev 1.3	Document No Document 990422-001
------	----------------	------------------------------------

programming language similar to Java that the developer creates as part of a special URL access point. The file contains instructions that when fetched by the pager are converted to byte-code by the gateway, and then downloaded to the handheld unit.

- Initially only simple primitives, loop controls and variables will be supported in the program, but in time any of the current API calls on the device will be accessible. By using RIM's new advanced API calls on the Blackberry Handheld the program can do a full range of activities on the device. For example they could:
 - Download the program and have a new Ribbon item appear to reference their program
 - Create a new set of options in the Pager Configuration section
 - Download a new Message filter program and support a new Message API attachment type
 - Download an Internet Browser that can support special interpretive byte-codes
- With these advanced download methods the user needs the ability to protect what gets downloaded. There is a concern that a virus or a malicious program could be downloaded that could disrupt the Blackberry Handheld to the point where it could not be used.
- It is expected that in time RIM will be able to offer PKI (Public Key Infrastructure) support for companies sending native RIM-Content to a Blackberry Handheld. In the configuration the company would be adapting information for the Blackberry Handheld using a programming language like Java. RIM would offer a library to these companies that would get a public key for them, convert the Blackberry Java program to byte-codes, then sign the information turning it into binary content. Then when a handheld fetches the information they would get a signed binary file, which can be unsigned and executed as a byte-code program directly.
- By using this byte-code method it would be possible to convert the 'browser' on the device to a program interpreter. The program interpreter would execute the downloaded instructions and do whatever the end-developer instructed. To achieve the necessary browser functionality discussed earlier, the device would simulate the look and feel of the browser. This would take place by downloading programs that embellish the contents of a given web page, possibly even simulate the appearance of an HTML form on the Blackberry Handheld.
- By using this proprietary byte-code method RIM can have the option of supporting some or all of the following :
 - HTML and HTML Form tags
 - XML (WML Specific dictionary)
 - WML Scripting
 - Visual Basic Scripting
 - HDML 2.0 and relating scripting
 - Java and other special embedded Java versionsThese advanced formats and languages could all be translated into the RIM proprietary byte code. The Blackberry byte code could be kept confidential to ensure that it can change in the future if necessary.



Date	Rev Rev 1.3	Document No Document 990422-001
------	----------------	------------------------------------

5 Functional Details

5.1 Message Viewer

The Message Viewer has very few changes necessary in order to achieve the direct link between the e-mail message and the URL link. The basic e-mail viewer would present the URL link in one of several ways as shown in below (initially it is expected the second two formats will be the methods used):

From: Sender (Email) Subject: This is a test Message
This is a URL Link: HTTP:// Blackberry/Sample/Link the user can roll to this link and double click or press enter.

From: Sender (Email) Subject: This is a test Message
This is a URL Link: <u>HTTP:// Blackberry/Sample/Link</u> the user can roll to this link and double click or press enter.

From: Sender (Email) Subject: This is a test Message
This is a URL Link: HTTP://Blackberry/Sample/Link the user can roll to this link and double click or press enter.

From: Sender (Email) Subject: This is a test Message
This is a longer Link: HTTP://Blackberry/Sample/special/ Link the user can roll to this link and double

Once the message is received the user will see all URL links as either bolded, highlighted, or presented on their own line, depending on what is possible on the target device. Naturally the goal is to keep both the current BlackBerry Handheld in mind and the upcoming Proton HandHeld.

When the cursor is sitting on any character of the URL link the hot key for jumping to the link will be the enter key. Otherwise if the user presses the roller wheel while on any URL link and a new default menu item will appear as shown below:

In the case where the user is sitting on a URL link the default is the command Follow the Link. This is similar to attachment processing when the user is sitting on an attachment and the 'Open Attachment' menu option is the default.

From: Sender	Save Message
Subject: This is	Reply to Message
This is a URL l	Reply to All
Blackberry/Sa	Forward Message
roll to this link	Replay With Text
enter.	Follow the Link

To identify the URL link the e-mail viewer will have to scan all message bodies and look for HTTP or WWW identifiers with other special characters like '//', '/' and '.'; but there are no guarantees and mistakes could be made. The goal will be for the program to follow the same rules as Outlook, which works in almost all cases.



6 Internet Browser

There are two possible designs for the Internet Browser, which offer very different functionality. The first is a traditional browser model where the program is nothing more than a thin HTML interpreter. The second design is not a browser at all but a file explorer and program interpreter. RIM has decided to develop the program interpreter that can behave like a browser. A review of the reasons for this choice can be found below.

6.1 Traditional Internet Browser

If RIM followed the Internet Browser model the user would be able to follow links, either from e-mail or the browser, but little more. It might be possible to support HTML-forms, to a limited degree, but scripting would be much more difficult to achieve. The other major limitation is that customization would be limited, and each time RIM changed the browser every device would have to be updated and new software deployed to the field.

The user would invoke URL after URL, using the cache where necessary, but little else. The Internet Browser would be pre-loaded on the device, probably downloaded via the Blackberry download procedure, and would not change. In time a scripting language would be bolted onto the browser and this would allow advanced forms and parameter verification. To create the scripting or forms component all pagers would have to be updated and possibility no revenue would be received from the update. The gateway would probably convert the wordy HTML forms into byte code for shipping over the wireless link. As these byte codes expanded more and more were added the user would have to update their device.

6.2 Program Interpreter

In this second design the new program is a Program Interpreter with two main roles. The first role is to present the available programs to be executed and allow them to maneuver through the files available; similar to viewing the cache. The second role is to load and execute the programs as requested by the user, or as fetched from a web site.

As a program viewer it simply displays the programs as a list of choices or as a hierarchy of programs in a manner that is similar to the message list box. The hierarchy can be initially established by the sender or determined by the user through menu options.

The diagram illustrates a presentation that is close to the message list. The first line has a title that shows coverage and pending events. Each sub-group has a floating title, like a date in the message list, and each group of program types. The floating titles could say anything that makes sense for the user.




Program Browser – Company Web... ✓	
Company Web Pages	
<input checked="" type="checkbox"/>	01/04 General Sales Listing
<input type="checkbox"/>	01/04 Sales Forecast Overview
FORMS – Scripted Forms	
<input checked="" type="checkbox"/>	03/04 Inventory Level



Practically speaking every item that can be selected is a program. Some programs can be viewed or perceived as HTML, but they are byte code instructions that simply calls print statements with strings. In the second case extra information apart of the program name could be shown, for example a file opened indicator could be present and a date field so that the user can see when it was last updated and whether a new copy is required.






When the user invokes the program they are given the ability to move through various available options and select which element they wish to run. All names presented are arbitrary and how these titles and relationships get setup is part of the downloading procedure, or set up by the user.

Within this list screen the user can move around between each item using the roller and a inverse video cursor highlights the current location. The user can either press enter or click the roller to get the menu for this screen. The enter is the usual hot key to save time.

Program Browser – Company Web... ✓	
Company Web Pages	
	01/04 General Sales Listing
	01/04 Sales Forecast Overview
Forms and Scripts	
	03/04 Inventory Level

If the user stops on a floating title they can perform a 'delete' command and delete the entire section. Where possible the functionality should be kept similar to the existing message list screen. At any time the user can change which title a given Gizmo is under and re-assign it from the default it was given when it was downloaded.

As the user gets the menu for this screen they are presented with a long list of menu choices. The most common choice is the 'Open Gizmo' menu item. At any time the user can select 'Saved Gizmos', which will save the current Gizmo value and place it into a cache of saved items for long-term storage. By selecting 'Move Gizmo' the user can re-assign the Gizmo to another title section. The new commands added are the 'Restrict Gizmo' and 'Unrestricted Gizmo', these are to add client side filtering. For full details see the BlackBerry Handheld Filtering section.

Program Brov	Hide Menu
Comp	Next Gizmo Title
	Delete Gizmo
	01/04 Ger
	01/04 Sal
Forms	Open Gizmo
	Move Gizmo
	03/04 Inv
	Previous Gizmo Title
	Refresh Gizmo
	Enter New Gizmo
	Restrict Gizmo
	Unrestricted Gizmo
	Save Gizmo

	Close

At any time the user can also "Enter a New Gizmo" address (URL value, Program Identifier, etc) and submit it for retrieval. This menu item will prompt the user for the input value to be fetched.

The user can also get accessed to saved Gizmos similar to saved messages. There will be a hot key and a menu item on the main ribbon screen that allows the user to quickly jump to the saved Gizmos folder.

Date	Rev Rev 1.3	Document No Document 990422-001
------	----------------	------------------------------------

When the user selects the move operation the screen and title changes to reflect a new list of items. This part of the browser is also used for title manipulation and creation. On this screen will be a listing of all titles defined thus far by the user or by programs sent over the air. The user can roll to a given title and make the move, or define a new title.

Move: Sales Forecast Overview...	✓
Public Web Sites	
Company Web Pages	
Forms and Scripts	
Personal Information	
Inventory Pages	

Once the user has rolled to the title they wish to manipulate they can press roller wheel to select to invoke the menu. If the user presses 'Enter' they will get the default, which would be to "Accept The Change" and move the item to the new title. However in this menu the user can also Add new Titles, Rename a given title and delete a title. Naturally the delete would prompt for confirmation since it will also delete program items under that title.

Move: Sales F	Hide Menu
Public Web	Deleted Gizmo
Company W	Accept Change
Forms and S	Add New Title
Personal Info	Rename Title
Inventory P	Delete Title
	Abort Move
	Close

When the user selects the item, by double clicking or by hitting enter, a 'loading' screen will be presented, shown in the illustration. The loading screen may only be presented for seconds, unless the program must be fetched first or if the program is very large.

Loading HTTP://file/location ...

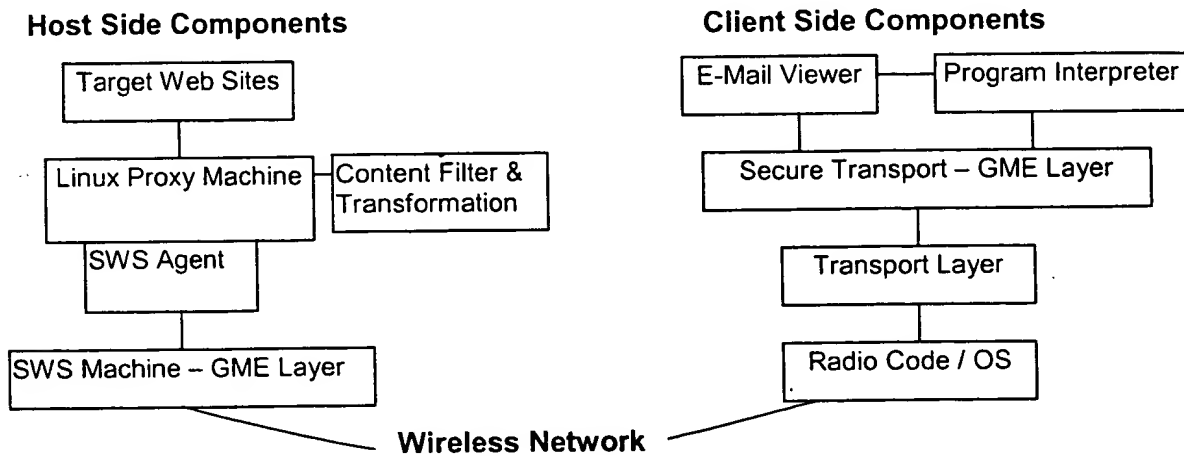
The loading screen is similar to the 'fetching' screen. The fetching screen appears when the interface is invoked from the message viewer, or when the user selects the 'Fetch Link' menu option. In this case the program has requested the URL and is waiting for it to be returned. If the user is fetching a random URL the program will prompt for the URL name before issuing the fetch. Once information arrives into the browser a notification tone is played following a user configured setting. This setting will be different from the message list so the user can determine if e-mail or Internet traffic has arrived.

Fetching HTTP://file/location ...

7 Host Development

7.1 RIM SWS Requirements

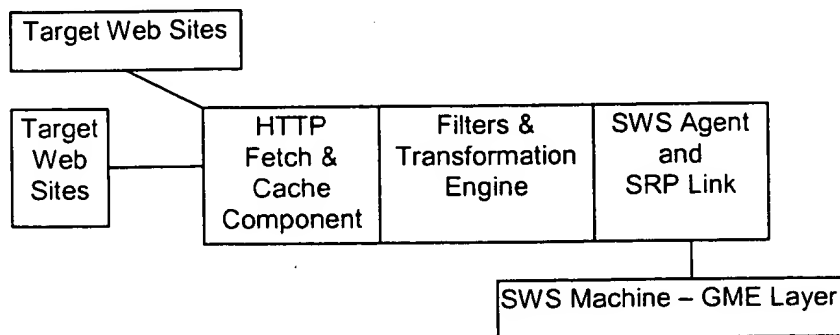
By using a simple gateway the requirements on the Host developers can be hidden unless they want the full information. The basic architecture is shown in the following illustration:



On the Host side of the connection there are several possible mechanisms to delivering information to the Blackberry Handheld. The method shown above uses a separate Linux Web Proxy to perform fetches on web pages. In this example the Content Filter could live within the Linux environment.

In this second example a single machine is used to both fetch information and perform filtering. This might be a first step method, as it requires less work and all the development can be focused on one machine.

Host Side Components

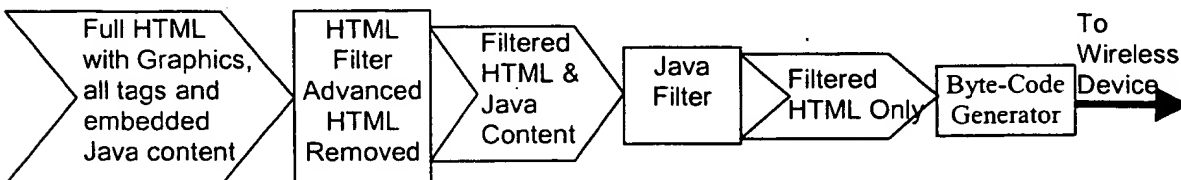


The filter and transformation engine simply strips off any HTML or scripting information that is not supported and then performs a byte-code transformation on the information so that it can be executed as a program on the device.

7.2 Host Filtering Requirements

On the host side of the connection, probably in close proximity to the SWS interface code, will be a series of filters and byte-code generators. The filters and byte-code generators will be 'linked in series' to create the necessary effect of limiting content to the device. The goal of the filters will be to take the raw HTML, WML, WML Scripting, Java or other content from the Internet and strip out any content that cannot be turned into byte-code instructions. In some cases more than one filter may have to be executed if the content includes different components, i.e. WML with WML Scripting embedded. Once this filtering has occurred the results are feed through a series of library calls to convert the information into RIM proprietary byte codes for execution on the device.

The figure below is an illustration of the process involved:



In this one example a given type of content is passed through a series of filters and byte-code generators until the final 'thin' content is sent to the wireless device. Ideally, within the filter sub-system these filters can be dynamically added or removed to solve a range of problems. The final stage is the byte-code generator that sends instructions to the handheld for execution.

7.3 Blackberry Handheld Filtering

It is expected that even with all the filtering that can be done on the Host side the user will still want additional text removed from the data sent to the handheld. This is because the average HTML web page has so much advertising, content, graphics and other related information that it cannot all be filtered automatically. This is where the new RIM Client filtering technique comes in to give the user control over further reduction in airtime traffic.

To limit the information further the user simple has to perform one of two possible steps, these include:

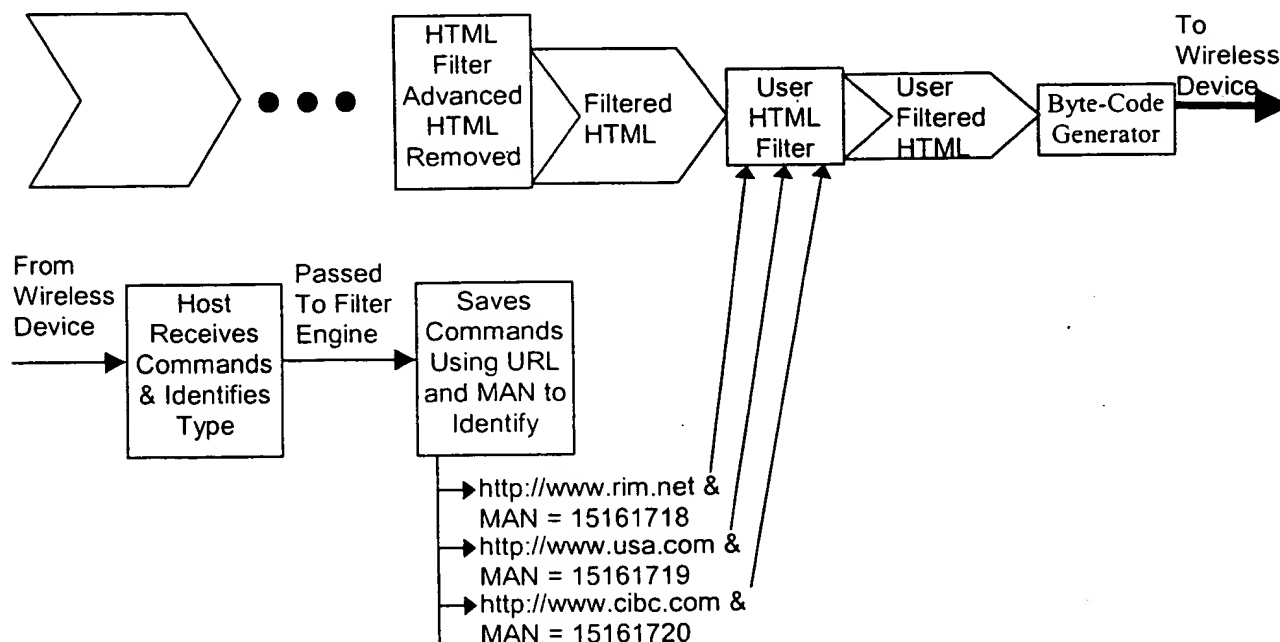
Date	Rev Rev 1.3	Document No Document 990422-001
------	----------------	------------------------------------

1. Marking the areas of text that is of interest to the user on the handheld and then sending a set of commands indicating those blocks back to the host.
2. Deleting the areas of text that are not of interest to the user on the handheld and then sending a set of commands indicating those blocks back to the host.

These two methods will be discussed in the patent application but for the rest of this document we will focus on the deletion method for restricting content to the device.

As part of an advanced filter on the host side the user can 'extend' it by sending additional filter commands and associating these commands to a specific URL address. By deleting unwanted text, the user can select the "Restrict Gizmo" command to send a set of commands to the host indicating what 'sections' of text they are interested in. If the user wishes to change his/her mind later they can select the "Unrestricted Gizmo" command to clear the filters at the host.

Following our previous example the following illustration demonstrates the dynamically increasing filter concept:



Due to space constraints as a users defines limiting filters on URLs that they receive they are returned back to the host for storage. Storage could either be a database or a regular file, saved with the URL name and device identifier (MAN or LLI). Then when the next time that page is fetched a quick attempt is made to read the file (URL.MAN) to retrieve further filtering commands.

7.4 Blackberry Filtering Commands

The commands used by Blackberry to filter content are based on the original information sent by the Proxy Server. In the creation of the original text each section of the HTML page, either each line or paragraph, will be given a tag or section identifier (Id).

During normal viewing of the message the user is simply moving the cursor across the displayed information as if they were viewing a mail message. Full link names are not shown, only small indicators to make the information more readable.

Fetches: Sales Forecast Overview	✓
The sales forecasts for the company appear to be projected to 1,000 units for the next 3 months. <Link>	
Previous sales for this same time last year were 3,000 units, showing an increase of 330% over a year	

If the user then invokes the menu they are given the option of marking the information they are not interested in, and restricting future fetches of the information. They can also follow a link, if the cursor is on a link, or expand or shrink link names for readability. It is expected that the user will rarely want the full link name and the information gives the user a context for the link to follow.

Fetches: Sales	Hide Menu
The sales for	Deleted Marked
appear to be	Start Marking
units for the	Send Restrictions
Previous sales	Unrestrict Information
last year were	Follow Link
an increase of	Full Link Names
	Short Link Names
	Close

When the user is in 'marking mode' the marked regions will jump in blocks, as defined by the original sections sent by the Proxy Server. In marking mode the user marks a section and presses the backspace key or 'Alt' Backspace key to remove the information. The can also press the 'Enter' key to delete the text or the backspace key since this is the default. After the user performs this on all sections they are not interested in, including any Links, they can select the menu and pick the "Send Restrictions" menu item. This will send a command with the deleted 'sections' to the Proxy Server for restriction next time. This also implies that the local cache will have to be updated or purged so that the next fetch doesn't get the wrong information. At any time while viewing a pager the user can also select the 'Unrestrict Information' menu item and remove any restriction commands that might exist on the Proxy Server.

Fetches: Sales Forecast Overview	✓
The sales forecasts for the company appear to be projected to 1,000 units for the next 3 months. <Link>	
Previous sales for this same time last year were 3,000 units, showing an increase of 330% over a year	



Date	Rev Rev 1.3	Document No Document 990422-001
------	----------------	------------------------------------

8 Content Provider

As mentioned in previous sections there are many types of content that can be delivered to the BlackBerry Handheld using the new Internet Content Proxy. Each of the content types are identified through HTTP and dealt with according to the filter requirements. One area that must be emphasized is that any given company has the ability to delivery 'native' RIM content to the BlackBerry Handheld for maximum flexibility. This option and other options are discussed below.

8.1 Native HTML Content

When sending native HTML content there are two forms that are expected. The first would be Internet Native, where the page was originally created for a full web browser, and the second would be BlackBerry enabled. The full HTML page is likely to be a disappointment to the user and it is expected that heavy filtering will be required at the proxy server. In the second case the web page creator can make certain adjustments to ensure the information sent to the device is minimized and sized to fit the devices screen limitations.

8.2 Native RIM Content

One of the most powerful aspects of running a byte-code interpreter on the BlackBerry handheld is that we can deliver programs to the pager for execution. The initial offering will allow 'partial' Java programs to be sent to the pager for execution. What parts of Java that will be allowed is still up in the air, but it will have the syntax and form of Java and RIM will not call it Java directly. This will allow a web page creator within any company to deliver working programs to the device for execution.

RIM will also add some minor extension to the language so that native BlackBerry API calls can be made. Each program that is downloaded can also specify a default 'category' it would like to be placed in. Refer back to the Program Interpreter section for details on this hierarchy browsing method.

The other major advantage of using RIM Native content is that RIM will provide a library to secure and sign the page before it is sent. This library would allow for a public key to be exchange and would convert the page to byte-codes before it is given to the Proxy Server. To perform this we still need a company to host the PKI server and a library needs to be distributed that does the byte-code conversion.



Date	Rev Rev 1.3	Document No Document 990422-001
------	----------------	------------------------------------

9 Conclusions

The RIM Internet content browser will not be a browser at all but a full featured program interpreter. Although the syntax and flexibility of the language used by this interpreter is not examined in this document, it will compare to most programming languages used today. This program interpreter will provided the 'effect' of the browser environment by turning HTML content into program interpretive commands. For example a page of HTML text turns into a series of 'PRINT' statements to the Blackberry Handheld client.

To manage a collection of loaded programs, or web pages stored as programs, the Command Interpreter has an 'explorer-like' interface for launching and fetching these items. In many cases the user does not know whether the items they are running are web pages or programs, since they exhibit features of both. The result is a HTML/Java experience on the pager and for the developer of content sent to the pager.